Junhua Yu 证明论 2017 - 第 1 课 - SOnion 笔记 (草稿)

snowonionlee@gmail.com

2018-10-15 18:25

本文期待的读者是啥样的人呢? 首先我自己是读者之一; 其次如果您完全不懂符号逻辑 (或者叫数理逻辑)(或简称逻辑), 肯定没法以本文来入门; 如果您学过一些逻辑, 可能能从本文得到各个逻辑的关系的梳理—也很可能从本文找出大大小小错误来, 不过我保证让一群人看到之前我至少会自己再三审读.

刻画逻辑的形式语言很重要, 解读它们的自然语言也很重要. 时间所限, 先只写核心的形式语言的东西, 起承转合的自然语言慢慢增补 (2018-10-11 14:54:29 我认为对我自己够用了).

我 LaTeX 不熟, 排版中文更不熟, 所以就请先看个影儿吧.

1 介绍

叫作 "证明论 (proof theory)" 的逻辑学分支有结构证明论、算术证明论、序数证明论等. 本课程讲的是结构证明论.

选课系统能看到的旧版课程简介 (局部):

证明论占据了逻辑学中的一个核心位置，主要研究形式演绎系统的一般规律。本课程全周上课，每周一次 3 课时，有期中和期末考试 (然而在 2017 年, 都没有. XD)。主要内容涵盖了公理系统、自然演绎系统、序列演算、归结等常用的证明论演算，并以经典命题逻辑、直觉主义命题逻辑、一阶逻辑、模态逻辑等重要的逻辑作为具体实例。这种安排将有利于学生在学习证明论的同时巩固和加深对这些重要逻辑的理解。此外，本课程还会介绍一些相关的专题来拓宽学生学术研究兴趣面。

2 预备知识

有的耳熟能详, 有的稍显陌生: 集合、多重集 (multiset)、序列; 多重集比集合多了 "出现次数", 序列比多重集多了 "出现次序". 图 (有向、无向)、树、在边上带标签的 (labeled) 树. Labeled finite tree 是许多结构的样子.

集合和多重集的区别很微妙. 应当指明一个 "看起来是集合的东西" 究竟是集合还是多重集, 以及 $\cap$、$\cup$ 等究竟是集合的交、并还是多重集的交、并.

本课程的重点是矢列演算 (sequent calculus). 在矢列演算 (的被本课程提到的部分) 里大量使用的是 "公式的多重集", 不是集合也不是序列.

"多重集是序列模掉 (modolo) 序 (order)."

3 当我们谈论 "一个逻辑"、"一个证明系统" 的时候在谈论什么

3.1 逻辑

我们在此约定: 本笔记系列谈论 "一个逻辑" 的时候, 说的是这个集合的一个元素:

$$\{\text{极小逻辑}, \text{直觉主义逻辑}, \text{古典逻辑}, \cdots\} \times \{\text{命题逻辑}, \text{一阶逻辑}, \text{模态逻辑}, \cdots\}$$

譬如 "古典命题逻辑" 是一个逻辑、"直觉主义模态逻辑" 是另一个逻辑.

同一个逻辑不一定总是 "长得一样". 有两个层次能让同一个逻辑的 "外貌" 发生变化. 以古典命题逻辑为例:

第一个层次, 选择不同的逻辑联结词集合, 该逻辑的公式的 (作为树的) 形态会不一样. 本笔记系列选取的联结词有 $\to, \wedge, \vee, \bot$. 其中, 蕴含、合取、析取是二元的, bottom 是零元的.

第二个层次, 对给定的逻辑联结词集合, 选择不同的具体语法, 该逻辑的公式的 (作为字符串的) 形态会不一样. 我们选取这个具体语法:

$$\phi ::= A \mid \bot \mid (\phi \to \phi) \mid (\phi \wedge \phi) \mid (\phi \vee \phi)$$

其中 A 代表原子命题. 我们常使用小写拉丁字母表示原子命题, 使用小写希腊字母表示公式.

并作以下约定:

1. 公式最外层的括号可以省略;
2. 优先级方面, 合取 > 析取 > 蕴含, 在此基础上若不引起歧义可以省略括号;
3. 结合性, 合取、析取、蕴含规定为右结合, 在此基础上若不引起歧义可以省略括号.

如: 把 $a \to b \to a$ 补全括号后是 $(a \to (b \to a))$, 把 $p \wedge q \wedge r \vee u \vee v \to w$ 补全括号之后是 $(((p \wedge (q \wedge r)) \vee (u \vee v)) \to w)$.

这样一来, 原子命题 p 是公式, $p \to \bot$ 是公式, $p \to$ 不是公式 (不符合具体语法), $\neg p$ 不是公式 (我们没有选取 $\neg$ 作为逻辑联结词!).

当我们需要引入 "否定" 的时候, 会把 $\neg\alpha$ 仅仅当成一个语法上的缩写, 将 $\neg\alpha$ 定义为 $\alpha \to \bot$. 也就是说, 写下 $\neg\alpha$ 的效果等同于写下 $\alpha \to \bot$ (并补充必要的括号). 值得指出, 在此种定义之下, $\neg$ 是元语言的符号, 而不是我们讨论的对象语言—命题逻辑的语言—的符号. 我们将会说 $\neg p$ 是公式—这其实是在说 $p \to \bot$ 是公式.

注:

- 事实上, 这个语法不仅适用于古典命题逻辑, 还适用于极小命题逻辑和直觉主义命题逻辑.
- 如果您有一些计算机科学, 特别是程序语言的知识, 可以看出: 第一个层次是抽象语法, 第二个层次是具体语法.

3.2 证明系统 **

读者大概已经知道, 讨论一个逻辑, 有两个截然不同的方面: 语义的和语形的. "真、假、赋值、解释、模型、永真式 (有效式)、可满足、不可满足、$\models$、$\Vdash$" 这些词汇和符号归属语义那边, "推导、公理、推理规则、定理 (内定理)、可证明、不可证明、$\vdash$" 则归属语形这边. 这是结构证明论的笔记, 因此讨论的几乎全都是语形的方面.

"一个证明系统" 关心一个逻辑的语形推导. 一个证明系统的 "类型"*** 是这个集合的一个元素:

{极小逻辑, 直觉主义逻辑, 古典逻辑 *, $\cdots$}×{命题逻辑, 一阶逻辑, 模态逻辑, $\cdots$}×{公理系统, 矢列演算, 自然演绎, $\cdots$}

极小、直觉主义、古典是逻辑的最常见的三种基底; 公理系统、矢列演算和自然演绎则是最常见的三类证明系统框架, 《Basic Proof Theory 第二版》称之为 "three types of formalism".

譬如我们在矢列演算的章节要详细谈的 `G1cp` , 就是古典命题逻辑的矢列演算系统的一种. c for classical (古典), p for propositional (命题), G for Gentzen (根岑, 矢列演算的发明人), 1 是个 "编号"—古典命题逻辑的矢列演算系统不止一种.

注:

- * 在本笔记系列中, "古典逻辑" 和 "经典逻辑" 是同义词, "一阶逻辑" 和 "一阶谓词逻辑" 是同义词.
- ** 在本笔记系列中, "证明系统"(proof system) 和 "演算"(calculus) 是同义词.
- *** 此处 "类型" 不是严格的术语!

我们紧接着要详细谈的是命题逻辑的公理系统.

4 希尔伯特风格的公理系统

在本笔记系列中, "公理系统"、"希尔伯特风格的公理系统"、"希尔伯特风格系统" 是同义词.

公理系统这种证明系统的特点是, 公理多, 规则少 (只有 MP 规则). 相比之下, 矢列演算的特点是公理少 (0 个), 规则多.

须知, 选取不同的公理集合, 也可能证明出同样的定理集合—也就是说形式化了同一个逻辑. 下文我们选取的是其中一种.

极小命题逻辑、直觉主义命题逻辑、古典命题逻辑中都只有 MP 规则, 而公理依次增多. 直觉主义是极小加上 “bottom 蕴含一切”, 古典是直觉主义加上双否消去或者加上排中律. 我们统一叙述 MP 规则, 然后依次叙述它们的公理.

MP 规则为:

$$\frac{\Gamma \vdash \alpha \qquad \Delta \vdash \alpha \to \beta}{\Gamma \cup \Delta \vdash \beta}\ (\mathrm{MP})$$

其中 α, β 为公式, Γ, Δ 为公式的集合, $\cup$ 是集合的并运算. $\vdash$ 是这样的二元关系: 以左边的公式集 (和该证明系统的公理) 为前提, 能推导出右边的公式. $\Gamma \vdash \alpha$ 读作 “以 Γ 为前提能推导出 α”, 也说 “α 是 Γ 的语法后承”. 当 Γ 为空集, 可以简写作 $\vdash \alpha$, 这时称 α 为该证明系统的定理.

于是, MP 规则的含义为: 如果以 Γ 为前提能推导出 α, 以 Δ 为前提能推导出 $\alpha \to \beta$, 那么从 “前提的总和” $\Gamma \cup \Delta$ 就能推导出 β.

是不是比 $\dfrac{\alpha \qquad \alpha \to \beta}{\beta}$ 要酷炫一些? :)

4.1 极小命题逻辑

采用 1.1 中定义的命题逻辑语言.

公理模式有 8 条:

1. $\vdash \alpha \to (\beta \to \alpha)$
2. $\vdash (\alpha \to (\beta \to \gamma)) \to ((\alpha \to \beta) \to (\alpha \to \gamma))$
3. $\vdash \alpha \to \alpha \vee \beta$
4. $\vdash \beta \to \alpha \vee \beta$
5. $\vdash (\alpha \to \gamma) \to ((\beta \to \gamma) \to ((\alpha \vee \beta) \to \gamma))$
6. $\vdash \alpha \wedge \beta \to \alpha$
7. $\vdash \alpha \wedge \beta \to \beta$
8. $\vdash \alpha \to (\beta \to (\alpha \wedge \beta))$

现在可以说一些注意事项了:

1. 注意到公理模式里没有 $\perp$ (因此也没有 $\neg$). 这导致, 极小命题逻辑里 $\perp$ 没有任何特殊性, 就是一个普通原子命题.
2. 什么是 “公理模式 (axoim schema)”? 因为其中的 α, β, γ 都是元语言符号 (更具体地, “元语言变元”). 将公理模式里的元语言变元替换成任意公式, 都能得到一个 (具体的) 公理. 这样子, 8 条公理模式对应着可数无穷条公理. 类似地, 我们也说 “定理模式”、“公式模式”.
3. 为什么一直说 “推导”(derive) 而避免说 “证明”(prove) 呢? 我用 “推导” 表示: 从公理和一些公式前提出发, 应用推理规则, 得到一个公式作为结论. “证明” 的用法则与《Basic Proof Theory 第二版》一致: Reserve “prove” in meta level, 即 “证明” 我一般指元定理 (譬如演绎定理、可靠性、完全性) 的证明.
4. 公理 3 和 4 是否重复? 不. 我们讨论语形推导, 不涉及语义, 所以 $\alpha \vee \beta$ 和 $\beta \vee \alpha$ 没什么关系.

我们来在极小命题逻辑里推导同一律 $\vdash \alpha \to \alpha$, 顺便展示公理系统的内定理的证明的树形的写法 (而非常见的序列形的写法).

Proof. .

$$\dfrac{\dfrac{\dfrac{}{\vdash \alpha \to ((\alpha \to \alpha) \to \alpha)}\ (\text{公理 } 1) \qquad \dfrac{}{\vdash (\alpha \to ((\alpha \to \alpha) \to \alpha)) \to ((\alpha \to (\alpha \to \alpha)) \to (\alpha \to \alpha))}\ (\text{公理 } 2)}{\vdash (\alpha \to (\alpha \to \alpha)) \to (\alpha \to \alpha)}\ (\mathrm{MP}) \qquad \dfrac{}{\vdash \alpha \to (\alpha \to \alpha)}}{\vdash \alpha \to \alpha}$$

□

(喷了! 突破天际了. 这可咋整⋯⋯)

注意到我们叙述的同一律也是个定理模式, 而非一个具体的定理. :D

4.2 直觉主义命题逻辑

在极小命题逻辑的基础上, 加一个公理模式, 即得到直觉主义命题逻辑:

9. $\vdash \bot \to \alpha$

加入公理 9 之后, $\bot$ 终于有了它的区别于其他原子命题的特殊性. 然后, 如 1.1 所提到的, 对于公式 α, 将 $\neg\alpha$ 定义为 $\alpha \to \bot$. 再次强调: 这是一个纯语法的替换; $\neg$ 是元语言的符号. 规定 $\neg$ 的优先级比 $\wedge$ 更高.

4.2.1 推不出的理

我们刚才用极小命题逻辑的公理模式和规则推导出了定理模式 $\alpha \to \alpha$, 于是我们说极小命题逻辑 (的这一个公理系统) 能推导出 $\alpha \to \alpha$. 而直觉主义命题逻辑加入公理 9 之后, 再利用公理 1 可推导出 $\alpha \to (\bot \to \alpha)$, 再利用公理 2 能推导出 $(\alpha \to \bot) \to (\alpha \to \alpha)$, 即 $\neg\alpha \to (\alpha \to \alpha)$. 这些都是 "能推导出" 的例子. 随着公理的增多, 能推导的定理也会变多.

那么, 直觉主义命题逻辑能否推导出这两个定理模式: $\vdash \neg\neg\alpha \to \alpha$ 和 $\vdash \alpha \vee \neg\alpha$ 呢? 答案是不能. "$\vdash \phi$ 不是证明系统 H 的内定理", 或者说 "H 不能 (仅从公理) 推导出公式 ϕ", 这是关于 H 的一个元定理, 或者是说 H 的一个性质; 它可以记作 $\nvdash_H \phi$.

一个公式 ϕ "不能被推导出 (underivability)" (或者为修辞而暂时不严格一下, "不可证") 是良定义的、可以证明的! 只需设法证明, 有限次地以任何方式使用公理与规则, 都不能推导出 ϕ, 即可. 在公理系统里证明 underivability 比较难, 在一些矢列演算里则相对简单.

4.3 古典命题逻辑

在直觉主义命题逻辑的基础上, 加以下两个公理模式之一, 即得到古典命题逻辑:

10. $\vdash \neg\neg\alpha \to \alpha$ (双否消去)

10. $\vdash \alpha \vee \neg\alpha$ (排中律)

4.3.1 谈可靠性和完全性, 要指定一对儿证明系统和语义

我们略微离题, 谈一谈逻辑的语义的方面. 读者可能知道, 把语义和语形联系起来的桥梁之一是可靠性 (soundness) 和完全性 (completeness). 同一个逻辑的证明系统可以有多种, 语义也可以有多种. 要谈论可靠性和完全性, 需要将证明系统和语义各指定一种.

古典命题逻辑的语义, 一般采用众所周知的那种: 赋值 V 是类型为 $P \to \{0,1\}$ 的函数, 其中 P 是原子命题的集合; 公式 ϕ 被 V 满足, 记作 $V \models \phi$, 其被递归定义为: $V \models p$ 当且仅当 $V(p) = 1$; $V \models \alpha \wedge \beta$ 当且仅当 $V \models \alpha$ 与 $V \models \beta$ 都成立; $V \models \alpha \vee \beta$ 当且仅当 $V \models \alpha$ 与 $V \models \beta$ 至少一个成立; $V \not\models \alpha \to \beta$ 当且仅当 $V \models \alpha$ 与 $V \not\models \beta$ 都成立; $V \not\models \bot$ 总是成立.

而直觉主义命题逻辑的语义就有各家所定义的很多种, 而且并不像古典逻辑的语义这么 "好操作". 极小命题逻辑的语义我没查过.

一阶逻辑和 (各种) 模态逻辑的语义又有各自的定义方式.

我们以叙述可靠性和完全性的定义作为结尾. 为此, 先定义语义后承: 公式 ϕ 是公式集合 C 的语义后承, 记作 $\Gamma \Vdash \phi$ (亦有文献使用 $\models$), 定义为: 对任意赋值 V, 若 [对任意 $\gamma \in \Gamma$ 有 $V \models \gamma$] 则 $V \models \phi$.

考虑一个逻辑的一个证明系统 C 和一个语义 S. (C, S) 这一对儿具有可靠性的定义为: 对任意公式集 Γ 和公式 ϕ, 若 (在 C 里有) $\Gamma \vdash \phi$ 则 (在 S 里有) $\Gamma \Vdash \phi$. 反过来, (C, S) 这一对儿具有完全性的定义为: 若 $\Gamma \Vdash \phi$ 则 $\Gamma \vdash \phi$.

4.4 读者练习

一个是证内定理, 一个是证元定理. 都用 2.1-2.3 给出的公理系统来做.

1. 在古典命题逻辑里推导定理模式 $\vdash (\neg\beta \to \neg\alpha) \to (\alpha \to \beta)$.

2. 记公式模式 $\eta = (((\alpha \to \beta) \to \alpha) \to \alpha)$. 请证明或证伪: 直觉主义命题逻辑加上 η 作为公理, 能推导出所有古典命题逻辑的定理 (这可以记作 $I_{pc} \oplus \eta \supseteq C_{pc}$).

5 版权、免责和负责声明

- 笔记有对授课内容的记录, 也有 (很多) 我自己的补充. 如果本文有错, 一般是我的理解、记录或打字错误, 而非 Junhua Yu 讲错 :D